

TDDD38/726G82 - Advanced programming in C++

Course Introduction

Christoffer Holm

Department of Computer and information science

- 1 What is TDDD38?
- 2 What is C++?
- 3 Very Basic C++

What is TDDD38?

Aim (syllabus)

- Explain non-trivial C++ language constructs and their semantics, e.g.
 - types, type conversion, temporary objects,
 - classes, derivation, polymorphism,
 - exception handling,
 - templates,
 - namespaces,
 - etc.

What is TDDD38?

Aim (syllabus)

- Explain the overall design principle of the C++ standard library, especially
 - the containers,
 - iterator,
 - algorithmsrelated parts.

What is TDDD38?

Aim (syllabus)

- Design and implement
 - usable,
 - correct,
 - error-safe,
 - non-trivial

classes and polymorphic classes.

What is TDDD38?

Aim (syllabus)

- Design and implement advanced program components, such as e.g.
 - traits classes,
 - policy classes,
 - function object classes

What is TDDD38?

Aim (syllabus)

- Use different components from the C++ standard library in combination to solve non-trivial computation problems, e.g. combine
 - standard containers algorithms,
 - iterators,
 - function objects,
 - own function objects,to design computations.

What is TDDD38?

Prerequisites (syllabus)

- Good knowledge and skills in programming
- Can use at least one procedural and/or object-oriented language (e.g. Ada, C, Pascal, Java or C++),
- knowledge about fundamentals of object-oriented programming (class, inheritance, polymorphism)

What is TDDD38?

Organization

- Self-study
- No mandatory hand-ins or labs
- Seminars
- Office hours
- E-mail: TDDD38@ida.liu.se
- Examination

What is TDDD38?

Extra information

- Course web page:
<http://www.ida.liu.se/~TDDD38/>
 - Planned seminars and material
 - Exercises
 - Useful links
 - Contact
 - Previous exams

What is TDDD38?

Extra information

- Course literature - optional
 - C++ Primer, 2012, Lippman, Lajoie, Moo
 - The C++ Programming Language, 4th edition, 2013, Stroustrup
 - A Tour of C++, 2013, Stroustrup
 - The C++ Standard Library, 2012, Josuttis
 - Effective Modern C++, 2014, Meyers
 - cppreference.com

What is TDDD38?

Evaluation from last term

Constructive feedback from students:

- Want mandatory labs
- Want recorded lectures

Response:

What is TDDD38?

Evaluation from last term

Constructive feedback from students:

- Want mandatory labs
- Want recorded lectures

Response:

- Mandatory labs are impossible due to limited resources.
This is *supposed* to be a self-study course.

What is TDDD38?

Evaluation from last term

Constructive feedback from students:

- Want mandatory labs
- Want recorded lectures

Response:

- Mandatory labs are impossible due to limited resources. This is *supposed* to be a self-study course.
- Lectures will *not* be recorded, due to several reasons (see course website for more information).

What is TDDD38?

Changes from last term

- More exercises
- More reading material
- More on C++20

What is TDDD38?

Topics

- Seminar 1
 - Data types, Initialization, Conversions
 - Value categories, Functions
 - Memory management & Pointers
- Seminar 2
- Seminar 3
- Seminar 4-6
- Seminar 7-?

What is TDDD38?

Topics

- Seminar 1
- Seminar 2
 - References
 - Classes
 - Lifetime Management
 - Operator Overloading
- Seminar 3
- Seminar 4-6
- Seminar 7-?

What is TDDD38?

Topics

- Seminar 1
- Seminar 2
- Seminar 3
 - Inheritance
 - Polymorphism
 - Exception Handling
 - Smart Pointers
- Seminar 4-6
- Seminar 7-?

What is TDDD38?

Topics

- Seminar 1
- Seminar 2
- Seminar 3
- Seminar 4-6
 - Function templates & class templates
 - Variadic templates, SFINAE
 - Compilation and Linking
 - Namespaces
- Seminar 7-?

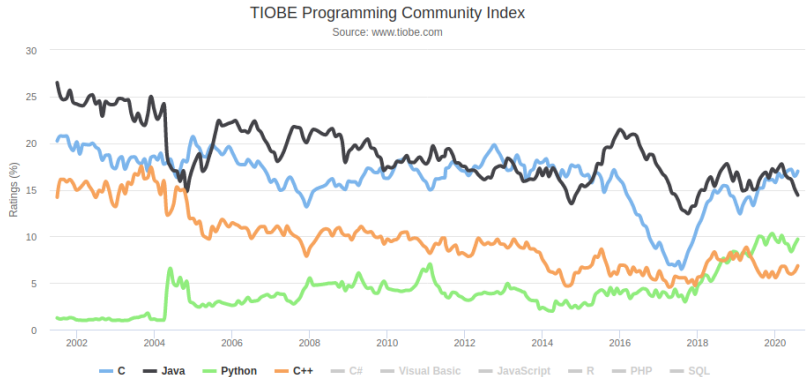
What is TDDD38?

Topics

- Seminar 1
- Seminar 2
- Seminar 3
- Seminar 4-6
- Seminar 7-?
 - Standard Template Library
 - Iterators, Containers, Function objects
 - Design patterns
 - Suggestions from you are welcome!

- 1 What is TDDD38?
- 2 **What is C++?**
- 3 Very Basic C++

What is C++?



What is C++?

Aug 2020	Aug 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.98%	+1.83%
2	1	▼	Java	14.43%	-1.60%
3	3		Python	9.69%	-0.33%
4	4		C++	6.84%	+0.78%
5	5		C#	4.68%	+0.83%
6	6		Visual Basic	4.66%	+0.97%
7	7		JavaScript	2.87%	+0.62%
8	20	▲	R	2.79%	+1.97%
9	8	▼	PHP	2.24%	+0.17%
10	10		SQL	1.46%	-0.17%

What is C++?

Definition

C++ is a general purpose programming language based on the C programming language [...]. In addition to the facilities provided by C, C++ provides additional data types, classes, templates, exceptions, namespaces, operator overloading, function name overloading, references, free store management operators, and additional library facilities

ISO Draft N4687 (C++17), §1/2

What is C++?

History

- 1979** First implementation of "C with classes"
- 1983** Rename to C++
- 1985** The C++ Programming Language first edition
- 1990** The Annotated C++ Reference Manual (ARM)
- 1998** First ISO standard (C++98)!
- 2003** Small amendments (C++03)
- 2011** C++11 standard! - The "new" C++
- 2014** Release of C++14
- 2017** Release of C++17
- 2020** C++20 standard! - Major release

What is C++?

General rules

- C++ must be useful *now*
- Support different styles
- Forcing the programmer is bad

What is C++?

General rules

- All features must be affordable
- Usefulness > misuse prevention
- Composition of different parts

What is C++?

Technical rules

- No implicit violations of the type system
- User-defines types = built-in types
- Locality is good
- When in doubt, choose the easiest alternative

What is C++?

Low-level rules

- Leave only assembler below C++
- Don't pay for what you don't use (zero-overhead)

What is C++?

Style

- No standardized style
- I will use a style, but you may use your own
- [Cpp Core Guidelines](#)
- This course will focus on modern C++ (C++14 and beyond)
- Usage of C features will be penalized

What is C++?

Important terms

- Implementation-defined behaviour
- Unspecified behaviour
- Undefined behaviour (UB)

- 1 What is TDDD38?
- 2 What is C++?
- 3 **Very Basic C++**

Very Basic C++

Hello World

```
#include <iostream>
int main()
{
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```

Very Basic C++

Bigger program

```
#include <iostream>
#include <string>
using std::cout;
using std::endl;
int main()
{
    cout << "What is your name? ";

    std::string name{};
    std::cin >> name;

    cout << "Your name is " << name << endl;
}
```

Very Basic C++

Bigger program

```
#include <iostream>
#include <string>
using std::cout;
using std::endl;
int main()
{
    int number{};
    cout << "Enter a number: ";
    std::cin >> number;
    if (number >= 0)
    {
        cout << "Your number is positive!" << endl;
    }
}
```

Very Basic C++

```
#include <iostream>
using namespace std;
int main()
{
    int counter{0}, sum{}, number{};
    cout << "Enter your numbers: ";
    while (counter < 5)
    {
        cin >> number;
        sum += number;
        ++counter;
    }
    cout << "The sum is: " << sum << endl;
}
```

Very Basic C++

```
#include <iostream>
using namespace std;
int main()
{
    int sum{}, number{};
    cout << "Enter your numbers: ";
    for (int i{0}; i < 5; ++i)
    {
        cin >> number;
        sum += number;
    }
    cout << "The sum is: " << sum << endl;
}
```

Very Basic C++

```
#include <iostream>
using namespace std;
int main()
{
    int number{};
    do
    {
        cout << "Enter number [0-10]: ";
        cin >> number;
    } while (number < 0 || number > 10);
}
```

Very Basic C++

```
#include <iostream>
using namespace std;
int main()
{
    int sum{}, number{};
    while (cin >> number)
    {
        sum += number;
    }
    cout << "The sum is: " << sum << endl;
}
```

www.liu.se